

(19) World Intellectual Property
Organization
International Bureau



INTERNATIONAL BUREAU OF PATENT COOPERATION
3, rue de la Harpe, 1205 Yvertois, Suisse
CH-1205 YVERTOIS, SUISSE
TEL. (41) 22 31 91 11
FAX (41) 22 31 91 21
E-MAIL: PCT@WIPO.ORG

(43) International Publication Date
12 May 2005 (12.05.2005)

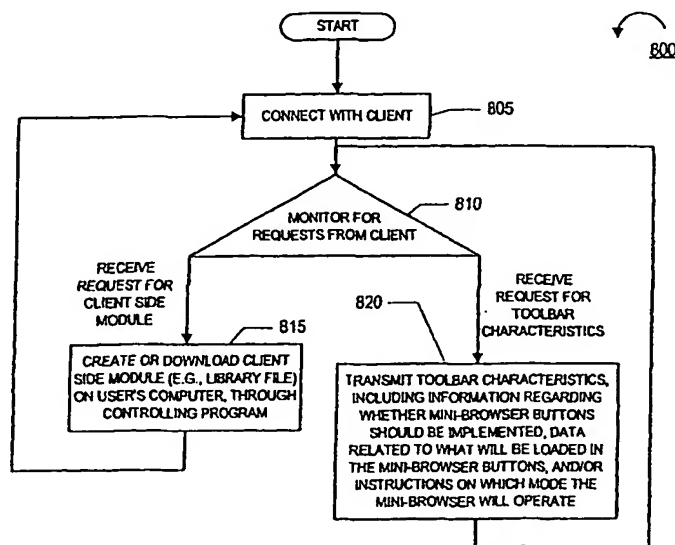
PCT

(10) International Publication Number
WO 2005/043355 A2

- (51) International Patent Classification⁷: G06F (74) Agents: DECARLO, James, J. et al.; Stroock & Stroock & Lavan LLP, 180 Maiden Lane, New York, NY 10038 (US).
- (21) International Application Number: PCT/US2004/037986
- (22) International Filing Date: 29 October 2004 (29.10.2004) (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/516,196 31 October 2003 (31.10.2003) US
10/794,705 5 March 2004 (05.03.2004) US
- (71) Applicant (for all designated States except US): YAHOO! INC. [US/US]; 701 First Avenue, Sunnyvale, CA 94089 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): WADA, Alan [US/US]; 861 Sladky Avenue, Mountain View, CA 94040 (US). BURKE, Eric [US/US]; 42404 Waterwheel Court, Northville, MI 48167 (US). FAN, Duke Tsao-Ming [US/US]; 1509 St. Francis Drive, San Jose, CA 95125 (US).
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND SYSTEM OF PROVIDING BROWSER FUNCTIONALITY THROUGH A BROWSER BUTTON



(57) Abstract: An embodiment of the invention provides for one or more miniature-browser buttons, used with a downloadable toolbar for an Internet browser. The method may comprise sending downloadable toolbar characteristics from a predetermined Internet site, which includes miniature-browser button characteristics, creating a miniature-browser button in the downloadable toolbar and loading the miniature-browser button using the miniature-browser button characteristics.

**METHOD AND SYSTEM OF PROVIDING BROWSER FUNCTIONALITY
THROUGH A BROWSER BUTTON**

CROSS-REFERENCE TO RELATED APPLICATION

[001] This application claims the benefit of U.S. Provisional Application Serial No. 60/516,196, filed October 31, 2003, entitled METHOD AND SYSTEM OF PROVIDING BROWSER FUNCTIONALITY THROUGH A BROWSER BUTTON, which is hereby incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[002] The invention is directed to computer interfaces and, more particularly, to a method and system of providing browser functionality through a button or toolbar of a browser.

2. Description of Related Art

[003] When accessing the Internet and the World Wide Web, an Internet user typically executes, via a computer, a browser software program such as, for example, Netscape Navigator™ or Microsoft™ Internet Explorer™. The browser program (i.e., a browser) establishes a link to the Internet (via a modem and an Internet Service Provider (ISP), for example) and also provides a textual and graphical user interface (for example, an application window), and included in the application window, a window for displaying Internet content (for example, a browser window) and toolbars for interfacing with the browser.

[004] The toolbars assist a user in interfacing with the browser. Toolbars can include textual regions, graphical regions, or a combination of both textual and graphical regions. Selecting an item on a toolbar by, for example, a mouse click, can instruct the browser to perform a certain function, or such selection may open a menu of items from which a user can choose from a plurality of items. Other browser toolbars allow a user to enter an Internet address and instruct the browser to navigate to that address. Some of the browser functions that can be implemented include, for example, save, forward, back, stop and refresh.

[005] A deficiency with present browser systems, however, is that the toolbars and their buttons are typically static in their look and functionality. Accordingly, a need exists for an improved method and system for toolbars and toolbar buttons with added functionality and richer content. In addition, although present browser systems allow users to make modifications to the toolbar settings, a deficiency with present browser systems is that the settings are limited to the computer where the modifications were implemented, and once set, are the same for each user who uses a computer. Accordingly, a need exists for an improved method and system for providing customizable buttons for a browser.

SUMMARY OF THE INVENTION

[006] The invention as claimed herein satisfies this and other needs, which will be apparent from the teachings herein. As mentioned above toolbar buttons and drop down menus are typically static in nature, including an image and/or text. In contrast, in accordance with an embodiment of the invention, dynamic content, which is displayable in a browser window, is created using any of the available tools used to create dynamic

and rich Web content, such as, for example Java, Javascript, flash, HTML, XML, etc. An appropriately sized browser window is displayed on a toolbar's buttons, menus, and/or menu items or may occupy a space where no button or interface element existed previously. The dynamic content is loaded into the browser window for display. Thus, the buttons, menus, and/or menu items or other elements of the browser user interface display dynamic content as opposed to only displaying static images and/or text. As used herein, the term miniature-browser button refers to a button or other toolbar or user interface element that can display content displayable by a browser.

[007] An embodiment of the invention includes one or more miniature-browsers buttons, used with a downloadable toolbar for an Internet browser. The method may comprise receiving downloadable toolbar characteristics, which include miniature-browser button characteristics, from a predetermined Internet site. Embodiments of the invention can also include creating a miniature-browser button in the downloadable toolbar and loading the miniature-browser button using the miniature-browser button characteristics.

[008] Other objects and features of the invention will become apparent from the following detailed description, considered in conjunction with the accompanying drawing figures. It is understood, however, that the drawings are designed solely for the purpose of illustration and not as a definition of the limits of the invention, for which reference should be made to the appended claims.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

- [0009] In the drawing figures, which are not to scale, and which are merely illustrative, and wherein like reference numerals depict like elements throughout the several views:
- [0010] FIG. 1 is a schematic illustrating an exemplary system implemented according to an embodiment of the invention;
- [0011] FIG. 2 is an exemplary screen shot illustrating a browser implemented according to an embodiment of the invention;
- [0012] FIG. 3 is an exemplary screen shot illustrating a browser toolbar implemented according to an embodiment of the invention;
- [0013] FIG. 4 is an exemplary screen shot illustrating a mouse pointer positioned over a miniature-browser button according to an embodiment of the invention;
- [0014] FIG. 5 is an exemplary screen shot illustrating a video display feature of a miniature-browser button according to an embodiment of the invention;
- [0015] FIG. 6 is an exemplary screen shot illustrating another video display feature of miniature-browser buttons according to an embodiment of the invention;
- [0016] FIG. 7 is an exemplary screen shot illustrating a browser toolbar including miniature-browser buttons implemented according to an embodiment of the invention;
- [0017] FIG. 8 is a flowchart illustrating an exemplary process of providing miniature-browser buttons according to an embodiment of the invention;
- [0018] FIG. 9 is a flowchart illustrating an exemplary process of providing and implementing miniature-browser buttons according to an embodiment of the invention;
- and

[0019] FIG. 10 is an exemplary screen shot illustrating a browser toolbar including miniature-browser buttons implemented according to an embodiment of the invention.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0020] There will now be shown and described in connection with the attached drawing figures several exemplary embodiments of a system and method of providing miniature-browser buttons.

[0021] With reference to FIG. 1, there is shown an exemplary block diagram of a system 100 implemented in accordance with certain embodiments of the invention. A computer 150 is connected to the Internet 190. While the Internet 190 is used in the exemplary embodiment of FIG. 1, the invention may also be used over any network of computers including local internets, intranets, local area networks (LANs), wide area networks (WANs), etc. Computer 150 includes an internal bus 164 that facilitates communication of information (i.e., digital data) between and among the various devices of the computer 150 and that also facilitates communication between the computer 150 and external devices and systems via a communication interface 168. A processor 166 coupled to the bus 164 processes information within the computer 150. The computer 150 also includes a memory 160 such as, for example, Random Access Memory (RAM) and/or other dynamic memory storage devices, coupled to bus 164 for receiving and storing instructions communicated from the processor 166. Memory 160 may also be used to temporarily store variable or other intermediate information while the processor 166 executes instructions. Read-Only-Memory (ROM) 162 is also coupled to the bus 164 for storing static data and instructions for use by the processor 166.

[0022] Various input and output devices are provided as part of computer 150, including, by way of non-limiting example, a display 154 (for example, cathode ray tube (CRT), liquid crystal display (LCD), etc.), an input device 156 such as a keyboard, and a cursor control device 158 such as a mouse, or trackball, for example. A data storage device 152 such as, for example, a magnetic disk drive and magnetic disk, a CD-ROM drive and CD-ROM, or other equivalent devices and data storage mediums, is coupled to the bus 164 for communication with the processor 166, main memory 160, and communication interface 168. The storage device 152 preferably has an operating system 170 and an Internet browser software program 172 (i.e., a browser) stored thereon. As will be discussed in greater detail below, a client-side module 174 may also be stored on the data storage device 152.

[0023] The computer 150 may communicatively connect to the Internet 190 via the communication interface 168 over one or more transmission media including, but not limited to, coaxial cable, copper wires, and fiber optical cables. Communication between the computer 150 and the Internet 190 may also be via a wireless medium, such as, for example, a cellular interface. The communication interface 168 facilitates two-way communication between the computer 150 and another electronic device or system, for example, a server computer or computers 126 provided by a content provider 120, 130.

[0024] An Internet user (not shown) using the computer 150 may gain access to the Internet 190 by causing the browser 172 to execute, thereby opening a communication link between the communication interface 168 of the computer 150 and an Internet site 126 of content provider 120, via an Internet Service Provider (ISP) 180. Internet content is communicated by the content provider 120 to the computer 150 for

display by browser 172. Alternatively, a content provider 120, 130 may also be an ISP 180.

[0025] In alternative embodiments, computer 150 may be a desktop or notebook computer, personal digital assistant (PDA), hand held device, or wireless phone (with graphics capability), or any other device now known or hereafter developed that is capable of performing the functions as described herein.

[0026] In accordance with an embodiment of the invention, a first Internet content provider 120 may provide an Internet user with access to a program 122 for controlling the browser 172. When executed by the user, the controlling program 122 downloads or creates a client-side module 174 such as, for example, a Dynamic Link Library (DLL), on the data storage device 152 of the Internet user's computer 150. The client-side module 174 preferably includes ActiveX control and/or Plug-in functionality. Thereafter, when the Internet user accesses the Internet using the browser 172, the browser 172 opens the client-side module 174 and preferably automatically establishes a connection to the content provider's Internet site 126. The content provider, in response to the connection established by the browser 172, loads information and/or functional data into a shell operating within the browser and created by the client-side module 174. For example, if the user has an account with the content provider 120, customized information and/or functionality may be loaded into the client-side module 174. If the user does not have an account, more generalized (for example, guest) information and/or functionality may be loaded. In an embodiment, information sent to the computer 150 is sent through a button feed from the content provider 120, as is described in further detail below.

[0027] The client-side module 174 opens a shell (or a plurality of shells) within the browser 172 that contains the ActiveX control or Plug-in code that may control, i.e., add to, remove from, and/or modify, the Internet browser 172. When loaded with the ActiveX control or Plug-in, the client-side module 174 preferably contains functions, objects, data, and other software, referred to generally herein as information, that may be used to control the browser 172. The client-side module 174 (and shell) can be persistent and thus, does not close when the Internet user moves, for example, from Internet site 126 (having a Web page 124) to Internet site 136 (having Web page 134). Thus, the information and/or functionality provided via the ActiveX control or Plug-in is not lost when the Internet user disconnects from the Internet site that loaded the ActiveX control or Plug-in, and connects to another Internet site. In alternate embodiments client-side module 174 may be located at a remote location from the Internet user.

[0028] An embodiment may be provided as a feature of a method of controlling an Internet browser interface displayable by an Internet browser on a display of a computer, and enabling a user of the computer and Internet browser to access and navigate the Internet and to receive and display on the computer one or more Web pages from one or more Internet sites, including the display of a Web page from a predetermined Internet site, the Internet browser having at least one Internet browser toolbar having at least one toolbar button providing a predetermined functionality to the user of the computer and Internet browser. The method can comprise providing, at the predetermined Internet site, access to a program for controlling the Internet browser interface and making available for downloading by the predetermined Internet site, a file for causing the display of a persistent user toolbar adjacent to the Internet browser toolbar

so as to create the visual impression that the user toolbar is an integrated part of the Internet browser, the user toolbar making additional functionality that is not part of the Internet browser prior to download available to the user after download as part of the Internet browser interface. Once the user toolbar is displayed the user toolbar remains displayed and the additional functionality remains available to the user regardless of a subsequent Internet site to which the Internet browser is caused to navigate after download. Such a method is described in further detail in U.S. Patent Application Serial No. 09/429,585, filed October 28, 1999, and entitled "A Method Of Controlling An Internet Browser Interface And A Controllable Browser Interface," the contents of which are hereby incorporated by reference herein.

[0029] A method of providing miniature-browser buttons may be implemented using, by way of non-limiting example, a client-side module 174 operating as a downloadable browser toolbar. An Internet user may customize the downloadable toolbar so that each time the user accesses the Internet using a browser, user-defined information and/or functionality will be displayed with the browser interface. The toolbar may include bookmarks, address and phone books, personal financial information, personalized news, and/or various functionality such as is available via ActiveX control and Plug-ins. When the toolbar is executed, the client-side module 174 can receive some, none or all of the toolbar buttons from a content provider 120. The toolbar buttons and their characteristics may be obtained through a feed from the content provider 120 to the client computer 150.

[0030] In addition, if an Internet user has an account with a content provider 120, that user's specific account information (for example, investment portfolio, news

headlines, bookmarks, address book, customized button, etc.) may be saved by the content provider 120 and dynamically displayed by different browsers on different computers. Also, two or more Internet users who share the same computer, but have different user accounts with a content provider 120 can each load their own account information, for example, pre-defined personal buttons, so that each user's personal buttons are displayed when that user logs into his or her respective account.

[0031] Downloadable browser toolbar buttons, menus and menu items are typically implemented using toolbar button drawing styles including a static icon/image and/or text. In embodiments of the invention, however, miniature-browser or mini-browser buttons are buttons on a toolbar which use a small Web browser control to display its interface. For a miniature-browser button, the outline of the button is drawn in a typical fashion, as is known in the art, but some or all of the content of the button is displayed using Web browser control(s). Therefore, dynamic and rich content that can be created for a Web page is made displayable in a button, menu and/or menu items as opposed to just displaying an image and/or text. The dynamic content may be created using HTML, DHTML, Flash, animated GIFs, Java, etc.

[0032] FIG. 2 is an exemplary screen shot illustrating a browser 200 implemented according to an embodiment of the invention. Browser 200 includes a plurality of windows that provide various functionalities to an Internet user. The browser 200 may comprise a first application window 205 that typically defines the general size, color, and layout of the browser 200 and includes window control buttons 270 (for example, minimize, close, etc.) for that application window 205. The browser 200 may also comprise a browser window 215 and toolbar windows 210. The browser window 215

and the toolbar windows 210 typically define information and/or functionality that will assist an Internet user when accessing and navigating the Internet. For example, the browser window 215 and the toolbar windows 210 may provide toolbars, pull-down menus, Plug-ins, applications, etc. A browser window 215 is provided as part of the browser 200 within which content from an Internet content provider 120 (see FIG. 1) may be displayed.

[0033] Toolbar windows 210 provided at the top (see FIG. 2) of the application window 205 define four toolbars 225, 230, 235, 240, which may include a variety of interface controls such as, for example, pull-down menus, functional buttons (for example, stop, back, forward, home, etc.), and a combination of functional buttons and windows (for example, a search button and window). The upper most toolbar 225, also known as a menu bar, provides a plurality of pull-down menus; the second toolbar from the top 230 provides a plurality of functional buttons; the third toolbar from the top 235 provides a pull-down menu and a window, for example, a Universal Resource Identifier (URI) address window; the bottom most toolbar 240 is executed by the client-side module 174 and provides added functionality to the browser separate from that provided by the browser prior to download of the client-side module 174, such as a search field, a movie trailer button 245 and an information scroll button 250. Buttons 245 and 250 are implemented using miniature-browser buttons as described below.

[0034] FIG. 3 is a screenshot illustrating an embodiment of the bottom most toolbar 240 implemented in accordance with the invention. Toolbar 240 can include one or more of a shortcut button to a content provider 315, such as, for example, a shortcut to Yahoo!, a toolbar settings button 320, a search region 330, which includes a search field

335 and a search button 340, a movie trailer button 245, an information scroll button 250 and a button 355 that creates a drop down menu including more toolbar items for an Internet user. These additional items may include a shortcut to e-mail and shopping functions (not shown). Movie trailer button 245 and a information scroll button 250 are exemplary miniature-browser buttons implemented in accordance with an embodiment of the invention.

[0035] Toolbar 240 can be implemented using one or more of three categories of buttons. The first category of button is called BUTTON. A BUTTON button performs an action when any part of the button is selected (for example, clicked on). Shortcut button 315 is an example of a BUTTON button. Clicking on any part of button 315 will navigate the browser 200 to a particular content provider's Web page. BUTTON buttons, such as button 315 can be implemented using typical methods, for example by displaying an image and/or text or as a miniature-browser button, where a browser window capable of displaying a variety of Internet content is used to display a button's content. An Internet provider can manipulate the URI assigned to the shortcut button through a button feed.

[0036] The second category of button is called MENU. When the user clicks anywhere on a MENU button, a menu including additional options is displayed. For example clicking on any part of toolbar settings button 320 causes a menu to appear below the button 320. The menu includes a list of additional options such as, for example, adding and/or editing toolbar buttons, changing the layout of the toolbar and refreshing the toolbar. Some menu items may open a second menu including more items. An Internet content provider can use a button feed to define the number of items that are

in a menu. The main MENU button itself or any or all menu items may be implemented as miniature-browser buttons. For example, the main MENU button and the first two items on the menu can be implemented as a miniature-browser buttons, while the remaining items on the menu are implemented in a typical manner.

[0037] The third category of button is called BUTTONMENU. BUTTONMENU buttons include a clickable region of the button that performs an action, and a clickable region that displays a menu. Search button 340 is an example of a BUTTONMENU button. Selecting the "Search" portion of the button 340 initiates a search for the term entered in the search field 335, while selecting the arrow portion of the search button 340 causes a menu to appear below the button with additional options for the user. The main BUTTONMENU button and/or any or all menu items can be implemented as miniature-browser buttons. For example, one button can have the main region of the button implemented in a typical fashion, but a BUTTONMENU button can include a menu with two menu items, the first being implemented as a miniature-browser button and the second being implemented as a typical menu item.

[0038] Movie trailer button 245 and information scroll button 250 are two non-limiting examples of the functionality that can be added to a toolbar button. FIG. 4 illustrates the movie trailer button when a user positions a mouse cursor over the button. As with typical toolbar buttons, a border is formed around the button 245 to indicate to the user which button on the toolbar will activate upon selection.

[0039] Movie trailer button 245 is a MENU button. As illustrated in FIG. 5, when a user selects button 245, a menu 505 including a video player 510 is displayed under the button. The video may be a streaming video offered by any Internet content

provider and can be played using a plug-in such as, for example, Windows Media™ Player. In addition, as illustrated in FIG. 6, selecting (for example, clicking) menu 505 creates a second browser window 605, which includes the video player 510. In alternate embodiments, a second browser window may be created by clicking on the video player 510 or by selecting a menu item, included in addition to the video player 510, for creating a second browser window. While previous menu items included static icons and/or text, the invention allows for the implementation of a menu item that displays video. Additional functionality may be added to the menu using different browser related applications such as, for example, applets and/or scripts.

[0040] Information scroll button 250 is a BUTTON button that scrolls current weather conditions. FIGs. 2-8 illustrate information scroll button 250 at different display stages. The information scroll button 250 offers numerous improvements over a typical button because a greater amount of information can be displayed in the same amount of space as a typical button. In alternate embodiments the scrolling information may include stock prices, breaking news, or in certain embodiments the button can scroll a decorative design. As illustrated in FIG. 7, when a user selects button 250, the browser is navigated to a weather related Web page. In alternate embodiments a user could be directed to a Web page that is related to the information that is being scrolled. BUTTON buttons are not limited to the scrolling embodiment of button 250; they can be modified to include other browser related media, such as, for example applets and/or scripts. In addition, MENU or BUTTONMENU buttons are also capable of displaying scrolling information.

[0041] The invention is not limited to movie trailer buttons 245 and information scroll buttons 250. As illustrated in exemplary browser 200' of FIG. 10, any content that can be displayed in a browser window is displayable in a miniature-browser button. For example button content 245' may display a music player and selecting the button 245' opens menu 505', which displays content 510'. Content 510' may include a user interface for the music player, offering functionality such as loading new music, switching between play-lists, controlling the volume and/or equalization. In addition, button 250' may display an animated advertisement that attracts the attention of a user. Further implementations are also possible.

[0042] FIG. 8 illustrates an exemplary embodiment of a method 800 for a server side process for providing and implementing the invention. In an embodiment of the invention, a user desiring miniature-browser buttons would obtain, by, for example, downloading a client side module 174 from a content provider 120. This downloading can include the user accessing a content provider's Web page using an Internet browser. Therefore, processing of method 800 begins in step 805 with the content provider 120 connecting with a client computer 150, which may occur, for example, in response to a user visiting the content provider's Web page. Content provider 120 may connect with client computers 150 through a server 126. The server 126 may then provide the client computer 150 with, for example, a Web page including a number of different options such as creating a new account, signing into an existing account, and downloading a client-side module 174.

[0043] Processing proceeds from step 805 to step 810, where the server 126 monitors for requests from the client 150. When a request for a client-side module 174 is

received, processing proceeds from step 810 to step 815. In step 815, through the controlling program 122, the server 126 creates or uploads a client-side module 174 to the user's computer 150. The client-side module 174 may be, for example, a library file implementing a downloadable toolbar, such as toolbar 240. In some embodiments, an installation process is initiated to install the toolbar in the user's computer 150.

[0044] After the toolbar is obtained by the client 150, and the toolbar is executed, for example, as part of the installation process of the toolbar or in response to the user subsequently executing a browser application, in an embodiment of the invention, the toolbar automatically (or otherwise) connects with a content provider 120. Thus, method 800 proceeds from step 815 to step 805 where the content provider 120 connects with the client 150. The content provider can be content provider 120, from which client-side module 174 is downloaded, or alternatively, the content provider can be another site, for example, content provider 130.

[0045] Processing proceeds from step 805 to step 810 where the server 126 monitors for requests from the client 150. When a request for toolbar characteristics is received, processing proceeds from step 810 to step 820, where the server 126 transmits the toolbar characteristics to the client 150. The request may be an explicit request for toolbar characteristics and/or a content provider may automatically send toolbar characteristics in response to a general request for toolbar content when client-side module 174 is initiated. In an embodiment, the toolbar characteristics are sent to the computer 150 through a button feed. The feed includes information regarding whether miniature-browser buttons should be implemented, data related to what will be loaded in

the miniature-browser buttons, and instructions on which mode the miniature-browser will operate.

[0046] In certain embodiments, before the request for toolbar characteristics is received by the server 126, the user at client 150 can sign into or open a new account with the content provider 120. Having an account with the content provider 120 and allowing the content provider 120 to store button characteristics allows multiple users who use the same computer to each have personalized miniature-browser buttons when using the same computer. Another benefit of certain embodiments is that the same user can use his or her personalized button characteristics at different computers without having to re-set those characteristics at the different computers. Alternatively, if the client does not have an account with the content provider 120 or the client is not signed in, the content provider 120 may transmit standard or generic button characteristics. Alternatively, certain characteristics may be transmitted based upon the URI and/or ISP from which the user is accessing the Web site. Following step 820, processing returns to step 810, where the server 126 monitors for requests from the client 150. In some embodiments, the server 126 continues to monitor for requests from the client 150 until the client 150 signs out. The server 126 can also transmit toolbar characteristics to a user 150 to, for example, change the scrolling information or to alter the video displayed by a button, without receiving a request from the user.

[0047] FIG. 9 illustrates an exemplary embodiment of a method 900 for providing and implementing miniature-browser buttons in accordance with the invention. In this embodiment the miniature-browser buttons are part of a downloadable toolbar 240 for a browser 200. The method 900 is initiated, for example, by client-side module 174, which

was initiated with the initiation of a Web browser. The method starts in step 905, with the client 150 transmitting a request to a predetermined Internet site for toolbar characteristics. The request may be a generic request for toolbar characteristics for a particular user.

[0048] Processing proceeds from step 905 to step 910 where the client 150 receives toolbar characteristics from the predetermined Internet site, for example, as part of a button feed. The button feed includes instructions to the toolbar on how to display its buttons, menus and menu items. Since the downloadable toolbar can be customized to different users, instructions included in button feeds vary depending on the user of the toolbar. Therefore, the button feed includes parameters for rendering its buttons including, for example, the size, location, and/or category of the buttons and the text and images that is displayed in the buttons. Typically, the toolbar renders the buttons according to the instructions and displays the text and/or image received for the buttons, but according to an embodiment of the invention, the button feed can also include information directing the toolbar to render some or all of its buttons as miniature-browser buttons. In such embodiment, the button feed may also include a URI, which points to miniature-browser button data to be displayed in the button, and/or the button feed may include the data (for example, HTML, scripts, applets, etc.) itself. Instructions to the toolbar may be in the form of flags, numeric representations for instructions, executable computer code instructions, etc. In addition, the toolbar may determine what to display and in what mode depending on the data it receives. For example, if the toolbar receives text and an image for a button, then the button is rendered in a typical fashion, while if

the toolbar receives certain HTML code, the button is rendered as a miniature-browser button.

[0049] If the feed does not include instruction to use miniature-browser buttons, the toolbar renders its buttons in the typical manner, for example by displaying the text and/or image received in the button feed for the button. If the button feed does include instructions to implement miniature-browser buttons, processing proceeds from step 910 to 912, where the toolbar renders some or all of its buttons as miniature-browser buttons. In one exemplary embodiment, the style and extended style fields in a button feed to the client's computer 150 indicate which buttons should be rendered using miniature-browser buttons. The toolbar renders miniature-browser buttons by creating an embedded ActiveX control of type CLSID_WebBrowser and activating it in-place in the toolbar using standard object linking and embedding (OLE) practices, or other such practices, now known, or hereafter developed, by those skilled in the art, as instructed by present disclosure.

[0050] After the miniature-browser button is created, processing proceeds from step 912 to step 915 where the "drag and drop" functionality for the miniature-browser button is disabled. Current Internet browsers allow users to "drag and drop" hyperlinks, images, etc., located in a first browser window into a second browser window and thereby navigate the second browser window to the selected hyperlink, image, or other valid drop item. This feature can be disabled in miniature-browser buttons, so a user cannot inadvertently change the content displayed in their miniature-browser buttons.

[0051] Following step 915, processing proceeds to step 920 where the toolbar determines how to load the content to be displayed in the miniature-browser button. In

this exemplary non-limiting embodiment, the toolbar either receives a URI that is associated with previously created button characteristics, or it receives the data that it loads into the miniature-browser button. This URI or data, in certain embodiments, can be included in the URL field of a button feed to the client's computer 150. The feed also includes information instructing the toolbar how to load content into its miniature-browser buttons. For example, the button feed may include a flag indicating a URI or data, or the toolbar can decide based on the content it receives.

[0052] If the toolbar determines to load content into a miniature-browser button by navigating to a URI, processing proceeds from step 920 to step 925 where the miniature-browser button is navigated to the received URI. In an embodiment of the invention, this is accomplished by calling `Navigate()` on the Web browser control. In certain embodiments, the upper left hand corner of a Web page residing at the received URI is displayed in the miniature-browser button. For this reason, the Web page residing at the URI is preferably designed and implemented to work in conjunction with the miniature-browser button, such that, for example, information on the Web page is visible at the miniature-browser button. The Web page residing at the URI can be implemented using, for example, HTML, DHTML, Flash, animated GIFs, Java, etc, and can resided at content provider 120 or at another Internet location, such as, for example, content provider 130. These tools can be used to implement the movie trailer button and the information scroll button described above. Then, processing proceeds from step 925 to step 945, where the toolbar determines the user-interaction mode of the miniature-browser buttons.

[0053] Returning to step 920, if the toolbar determines to load received data (for example, raw HTML) into a miniature-browser button, processing proceeds from step 920 to step 930. In step 930, the miniature-browser button is navigated to a quickly loading Web page, such as, for example, "about:blank." Step 930 is performed because certain functions are only available to a browser after it has fully loaded a page. In an embodiment of the invention when, a `DISPID_DOCUMENTCOMPLETE` event is received from the miniature-browser button, an `HTMLDocument` is valid and additional functions are available to the browser.

[0054] Processing proceeds from step 930 to step 935 where a stream object is created using the received data (for example, raw HTML). Then in step 940 the stream object is loaded into the miniature-browser button using functions made available after loading the quickly loading Web page. In an embodiment of the invention, the stream object is loaded using a `Load()` method of the `IPersistStreamInit` interface of the `HTMLDocument2` object. The received data may be created using, for example, HTML, DHTML, Flash, animated GIFs, Java, etc. Then, processing proceeds from step 940 to step 945, where the toolbar determines the user-interaction mode of the miniature-browser buttons.

[0055] Once the miniature-browser button is displaying HTML or other rich content as described in the previous steps, processing proceeds to step 945 where the user-interaction mode is determined. In an exemplary embodiment, miniature-browser buttons operate in one of at least two user-interaction modes: normal mode, and override mode. The button feed includes information instructing the toolbar which mode to

operate the miniature-browser buttons, in this example, the available modes are normal mode and override mode.

[0056] If the toolbar determines to operate the miniature-browser button in a normal mode, processing proceeds from step 945 to step 950, where the user's selections (for example mouse clicks) and mouse movements are handled by the miniature-browser button. Miniature-browser button events are not monitored or modified before being handled by the miniature-browser button. Then method 900 stops in step 970, for example, with the user logging out, or closing the browser.

[0057] Returning to step 945, if the toolbar determines to operate the miniature-browser button in an override mode, processing proceeds from step 945 to step 955, where the toolbar intercepts mouse events to the miniature-browser button and treats the miniature-browser button as though it were a typical toolbar button by, for example, preventing the mouse pointer from turning into a hand when it is positioned over a hyperlink displayed in a miniature-browser button. Thus, user interaction with the toolbar buttons remains continuous. In addition, URIs to which a miniature-browser button should navigate if selected, in override mode, are sent in a button feed similar to regular buttons. Therefore, the toolbar overrides and handles events, for example, navigating a user's main browser window to a URI associated with a shortcut button, otherwise handled by a miniature-browser button.

[0058] In step 955, the miniature-browser button is prepared to intercept and modify button event messages so that the toolbar can view a button's events before they are handled by the button. In an exemplary embodiment, messages are intercepted, as is known in the art, through window subclassing. The miniature-browser button is

subclassed at the time the first DISPID_BEFORENAVIGATE2 event is received, i.e., the first time the browser window is available.

[0059] Processing proceeds from step 955 to step 960, where the miniature-browser button monitors events before they are performed. In an embodiment, subclassing the miniature-browser button allows button down/up/double-click messages, mouse moves, timer events, and SETCURSOR events to be viewed before they are implemented. When an event is received, processing proceeds to step 965, where the event is modified as necessary. In an embodiment, mouse click and movement messages are passed to the parent window, which is the toolbar. This window then handles the event as though the user interacted with a typical toolbar button or menu item.

[0060] For example, in OVERRIDE mode the toolbar overrides the normal mouse appearance so that user mouse movements over the miniature-browser button do not change the cursor to a hand when it is over a link. In an embodiment, this is accomplished by subclassing all the miniature-browser's child windows and handling the WM_SETCURSOR message. When this message is intercepted the operating system, (for example, Windows) is notified that the event has been handled and the event does not get passed to the browser, and therefore the cursor does not change. Method 900 ends in step 970, for example with a user logging out or closing the browser window.

[0061] While the embodiment above describes miniature-browser buttons for adding rich content to a downloadable toolbar, the invention is not limited to this case. Additionally, miniature-browser button functionality may include advertisements with previews of new Web sites displayed in the button or menu, instant message windows loaded directly in a toolbar and Internet journal entry forms. Thus, while there have been

shown and described and pointed out fundamental novel features of the invention as applied to preferred embodiments thereof, it will be understood that various omissions and substitutions and changes in the form and details of the disclosed invention may be made by those skilled in the art without departing from the spirit of the invention. It is the intention, therefore, to be limited only as indicated by the scope of the claims appended hereto.

CLAIMS

What is claimed:

1. A method of providing buttons for a downloadable toolbar made available by an Internet content provider and used with an Internet browser having a browser interface, the method comprising:
 - sending downloadable toolbar characteristics from a predetermined Internet site, the toolbar characteristics including button characteristics for at least one button implemented as a miniature-browser button, wherein the miniature-browser button comprises an Internet browser display window, and the button characteristics include either of a universal resource identifier and loadable button content;
 - defining a region on the downloadable toolbar to display the miniature-browser button as part of the downloadable toolbar; and
 - loading the button characteristics to enable operation of the miniature-browser button.
2. The method of claim 1, wherein the button characteristics include both a universal resource identifier and loadable button content.
3. The method of claim 1, wherein loading the button characteristics comprises:

loading a quickly-loading Web page into the miniature-browser button;
creating a stream object using the loadable button content; and
loading the stream object into the miniature-browser button using
functions made available after loading the quickly-loading Web
page.

4. The method of claim 1, wherein the button characteristics include a user interaction mode for the miniature-browser button, wherein the miniature-browser button operates in either of a normal mode and an override mode.

5. The method of claim 4, wherein when the miniature-browser button operates in the normal mode, the method further comprises allowing the miniature-browser button to handle miniature-browser button events.

6. The method of claim 4, wherein when the miniature-browser button operates in the override mode, the method further comprises:
enabling the miniature-browser button to intercept and modify miniature-browser button events;
monitoring for a miniature-browser button event; and
modifying the event so that the miniature-browser button operates in a different manner than it otherwise would if not in override mode.

7. The method of claim 6 wherein the event includes changing a mouse pointer cursor to a hand icon when the mouse pointer cursor is positioned over a hyperlink in the miniature-browser button, and wherein the event is modified so that the mouse pointer cursor remains a mouse pointer cursor when placed over a hyperlink in the miniature-browser button.

8. The method of claim 6, wherein an event that would occur in a first manner when the miniature-browser button is in normal mode is modified such that the event occurs in a second manner, different from the first manner, when the miniature-button browser is in override mode.

9. The method of claim 6, wherein the enabling the miniature-browser button includes subclassing the miniature-browser button.

10. The method of claim 1, further comprising disabling at least one function of the miniature-browser button.

11. The method of claim 1, wherein the downloadable toolbar is implemented using an ActiveX control.

12. The method of claim 1, wherein the universal resource identifier represents an address on the Internet where content to be loaded in the miniature-browser button is located.

13. The method of claim 1, further comprising sending updated downloadable toolbar characteristics from the Internet content provider which includes updated loadable button content.

14. The method of claim 1, further comprising sending updated downloadable toolbar characteristics from the Internet content provider including universal resource identifiers that represent an address on the Internet where new content to be loaded in the miniature-browser button is located.

15. The method of claim 1, wherein the content provider maintains a user account comprising user account information.

16. The method of claim 15, further comprising providing the user with the ability to log in to the content provider.

17. The method of claim 16, wherein the user account information comprises downloadable toolbar button characteristics specific to the user.

18. The method of claim 17, wherein content of the downloadable toolbar button characteristics is determined by the content provider based upon the user account information such that the button characteristics of the miniature browser button are personalized in accordance with specific preferences of the user.

19. The method of claim 18, wherein the user account information is maintained by the content provider, independent of a first computer having a first browser that the user utilizes, such that button characteristics of the miniature browser button remain personalized when the user utilizes a second computer having a second browser.

20. The method of claim 1, wherein the button characteristics cause scrolling information content to be displayed within the miniature-browser button.

21. The method of claim 20, wherein the scrolling information content includes a hyperlink that may be activated by the user.

22. The method of claim 20, wherein the scrolling information content includes plural content portions, each of the content portions containing a hyperlink that may be activated by the user.

23. The method of claim 1, wherein the button characteristics identify loadable button content to be displayed within the miniature-browser button.

24. The method of claim 23, wherein the loadable button content includes Web page content to be displayed within the miniature-browser button.

25. The method of claim 23, wherein the loadable button content includes an instant messaging window to be displayed within the miniature-browser button.

26. The method of claim 23, wherein the loadable button content includes a blog to be displayed within the miniature-browser button.

27. The method of claim 23, wherein the loadable button content includes an advertisement to be displayed within the miniature-browser button.

28. The method of claim 23, wherein the loadable button content includes a scrolling message to be displayed within the miniature browser button.

29. The method of claim 28, wherein the scrolling message includes a hyperlink.

30. The method of claim 28, wherein the scrolling message includes plural portions, each of the portions containing a hyperlink that may be activated by the user.

31. The method of claim 1, wherein the loadable button content includes Web page content to be displayed within the miniature-browser button.

32. The method of claim 1, wherein the loadable button content includes an instant messaging window to be displayed within the miniature-browser button.
33. The method of claim 1, wherein the loadable button content includes a blog to be displayed within the miniature-browser button.
34. The method of claim 1, wherein the loadable button content includes an advertisement to be displayed within the miniature-browser button.
35. The method of claim 1, wherein the loadable button content includes a scrolling message to be displayed within the miniature browser button.
36. The method of claim 35, wherein the scrolling message includes a hyperlink.
37. The method of claim 35, wherein the scrolling message includes plural portions, each of the portions containing a hyperlink that may be activated by the user.
38. The method of claim 1, wherein a plurality of additional miniature-browser buttons are implemented in the downloadable toolbar.

39. The method of claim 1, wherein the downloadable toolbar is a floating toolbar capable of being positioned independently of and apart from the Internet browser interface.

40. The method of claim 1, wherein the downloadable toolbar is positionally anchored to the Internet browser interface.

41. The method of claim 1, wherein the toolbar is visually integrated into the Internet browser interface.

42. A method of providing personalized content to a user by an Internet content provider, the content being used with an Internet browser having a browser interface, the method comprising:

 sending downloadable toolbar characteristics from a predetermined

 Internet site, the toolbar characteristics including button

 characteristics for at least one button implemented as a miniature-

 browser button, wherein the miniature-browser button comprises

 an Internet browser display window, and the button characteristics

 include either of a universal resource identifier and loadable button content;

 defining a region on the downloadable toolbar to display the miniature-

 browser button as part of the downloadable toolbar;

loading the button characteristics to enable operation of the miniature-browser button;

wherein content of the downloadable toolbar button characteristics is determined by the content provider based upon user account information such that the button characteristics of the miniature browser button are personalized in accordance with preferences of the user; and

wherein the user account information is maintained by the content provider, independent of a first computer having a first browser that the user utilizes, such that button characteristics of the miniature browser button remain personalized when the user utilizes a second computer having a second browser.

43. The method of claim 42, wherein personalized Web page content is displayed within the miniature-browser button.

44. The method of claim 42, wherein a personalized instant messaging window is displayed within the miniature-browser button.

45. The method of claim 42, wherein a personalized blog is displayed within the miniature-browser button.

46. The method of claim 42, wherein an advertisement is displayed within the miniature-browser button.

47. The method of claim 46, wherein the advertisement is targeted to the user based upon the user account information.

48. The method of claim 42, wherein a personalized scrolling message is displayed within the miniature browser button.

49. The method of claim 48, wherein the scrolling message is targeted to the user based upon the user account information.

50. A method of advertising by providing buttons made available by an Internet content provider for a downloadable toolbar used with an Internet browser having a browser interface, the method comprising:

sending downloadable toolbar characteristics from a predetermined Internet site, the toolbar characteristics including button characteristics for at least one button implemented with a miniature-browser button, wherein the miniature-browser button includes an Internet browser display window, and the button characteristics include either of a universal resource identifier and loadable button content, the button content including advertisement content;

defining a region as part of the downloadable toolbar to display the miniature-browser button within the downloadable toolbar; and loading the button characteristics to enable operation of the miniature-browser button such that at least a portion of the advertisement content is displayed within the miniature-browser button.

51. A system for providing, from a server of an Internet content provider to a user computer, buttons used with an Internet browser having a browser interface, the system comprising:

one or more servers configured for:

sending downloadable toolbar characteristics from a predetermined Internet site, the toolbar characteristics including button characteristics for at least one button implemented as a miniature-browser button, wherein the miniature-browser button comprises an Internet browser display window, and the button characteristics include either of a universal resource identifier and loadable button content;

wherein a region is defined on the downloadable toolbar to display the miniature-browser button as part of the downloadable toolbar; and

button characteristics are loaded to enable operation of the miniature-browser button.

52. The system of claim 51, wherein the button characteristics include both a universal resource identifier and loadable button content.

53. The system of claim 51, wherein loading the button characteristics comprises:

loading a quickly-loading Web page into the miniature-browser button;
creating a stream object using the loadable button content; and
loading the stream object into the miniature-browser button using
functions made available after loading the quickly-loading Web
page.

54. The system of claim 51, wherein the button characteristics include a user interaction mode for the miniature-browser button, wherein the miniature-browser button operates in either of a normal mode and an override mode.

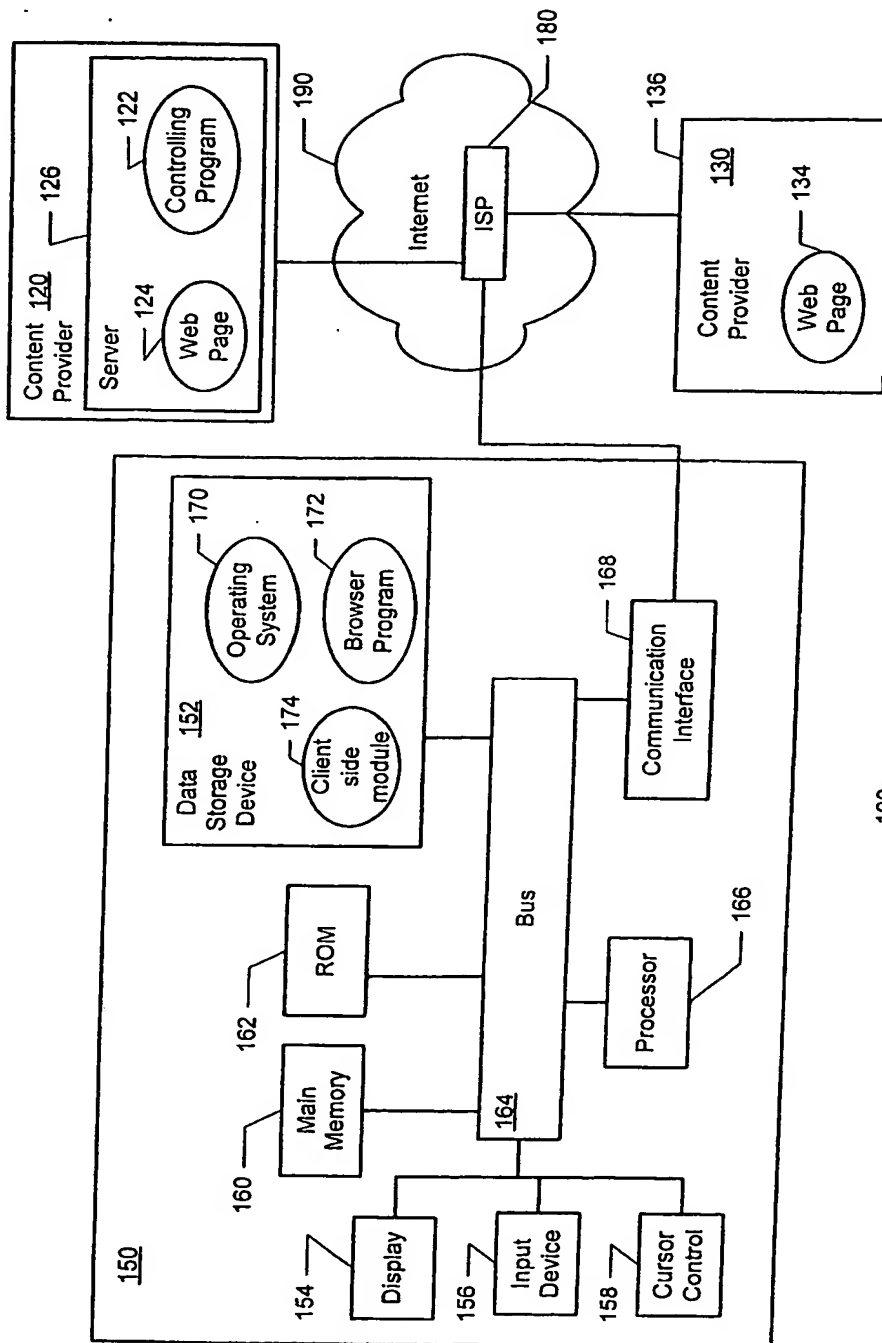
55. The system of claim 51, wherein the miniature-browser button operates in the normal mode, the method further comprising allowing the miniature-browser button to handle miniature-browser button events.

56. The system of claim 51, wherein when the miniature-browser button operates in the override mode, the server being further configured for:
enabling the miniature-browser button to intercept and modify miniature-browser button events;

monitoring for a miniature-browser button event; and
modifying the event so that the miniature-browser button operates in a
different manner than it otherwise would if not in override mode.

57. The system of claim 51, wherein the event includes changing a mouse pointer cursor to a hand icon when the mouse pointer cursor is positioned over a hyperlink in the miniature-browser button, and wherein the event is modified so that the mouse pointer cursor remains a mouse pointer cursor when placed over a hyperlink in the miniature-browser button.

58. The system of claim 51, wherein an event that would occur in a first manner when the miniature-browser button is in normal mode is modified such that the event occurs in a second manner, different from the first manner, when the miniature-button browser is in override mode.



100
FIG. 1

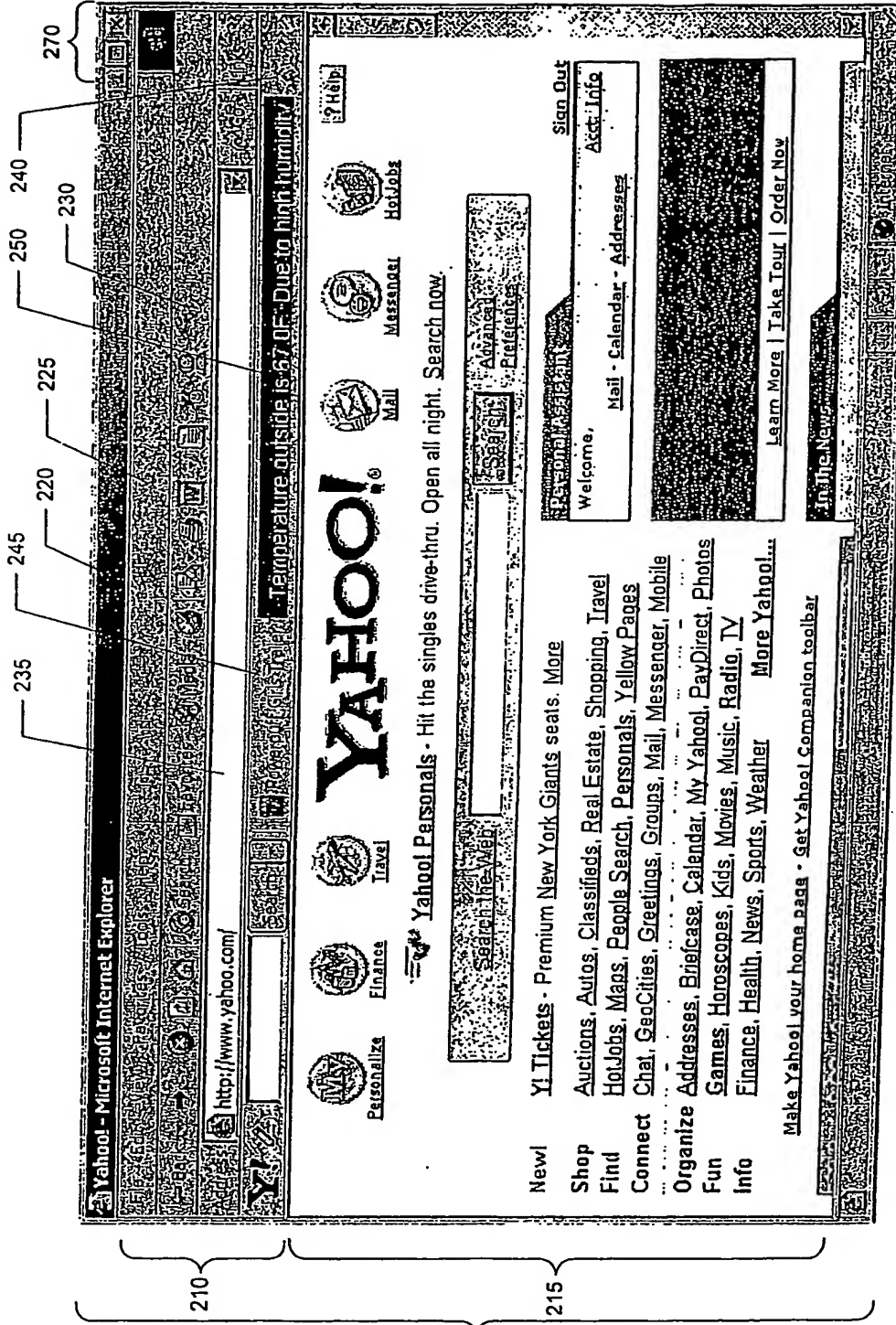


FIG. 2

200

BEST AVAILABLE COPY

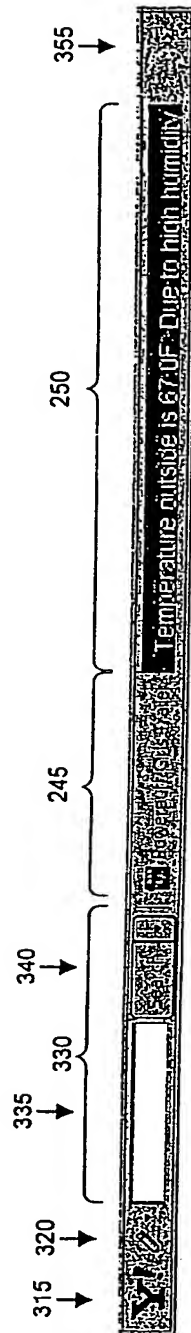


FIG. 3

240

BEST AVAILABLE COP

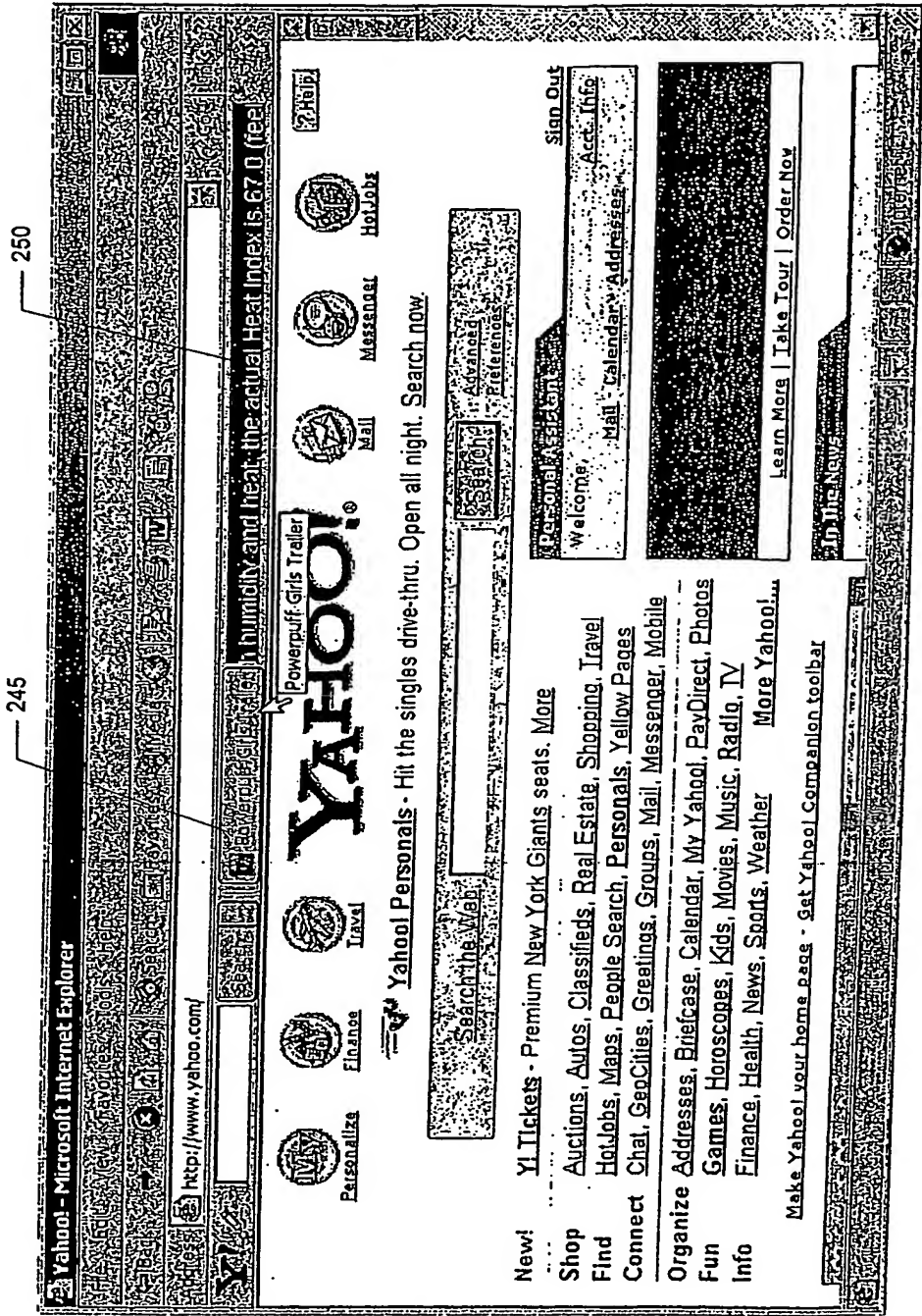


FIG. 4

BEST AVAILABLE COPY

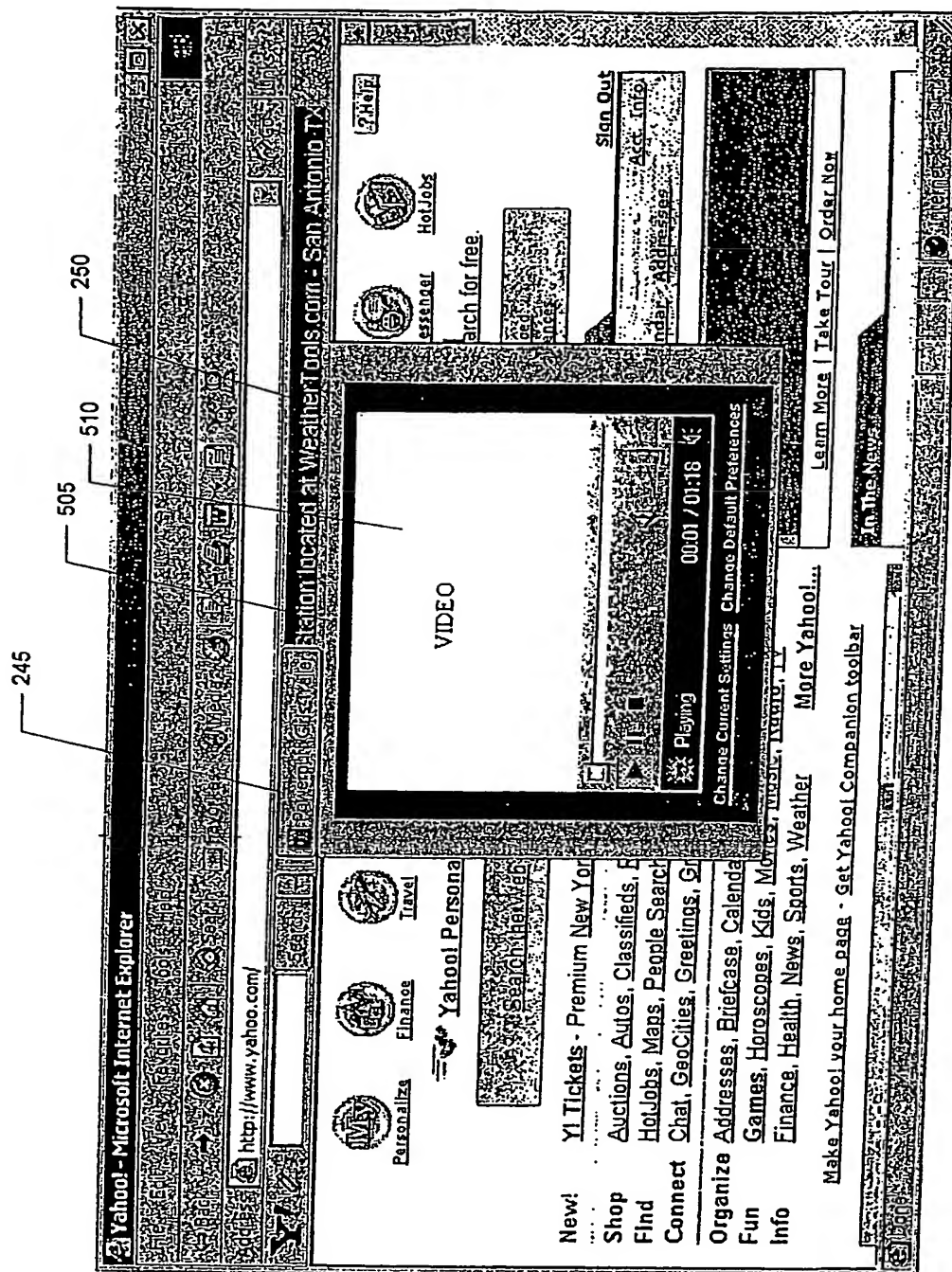


FIG. 5

BEST AVAILABLE COPY

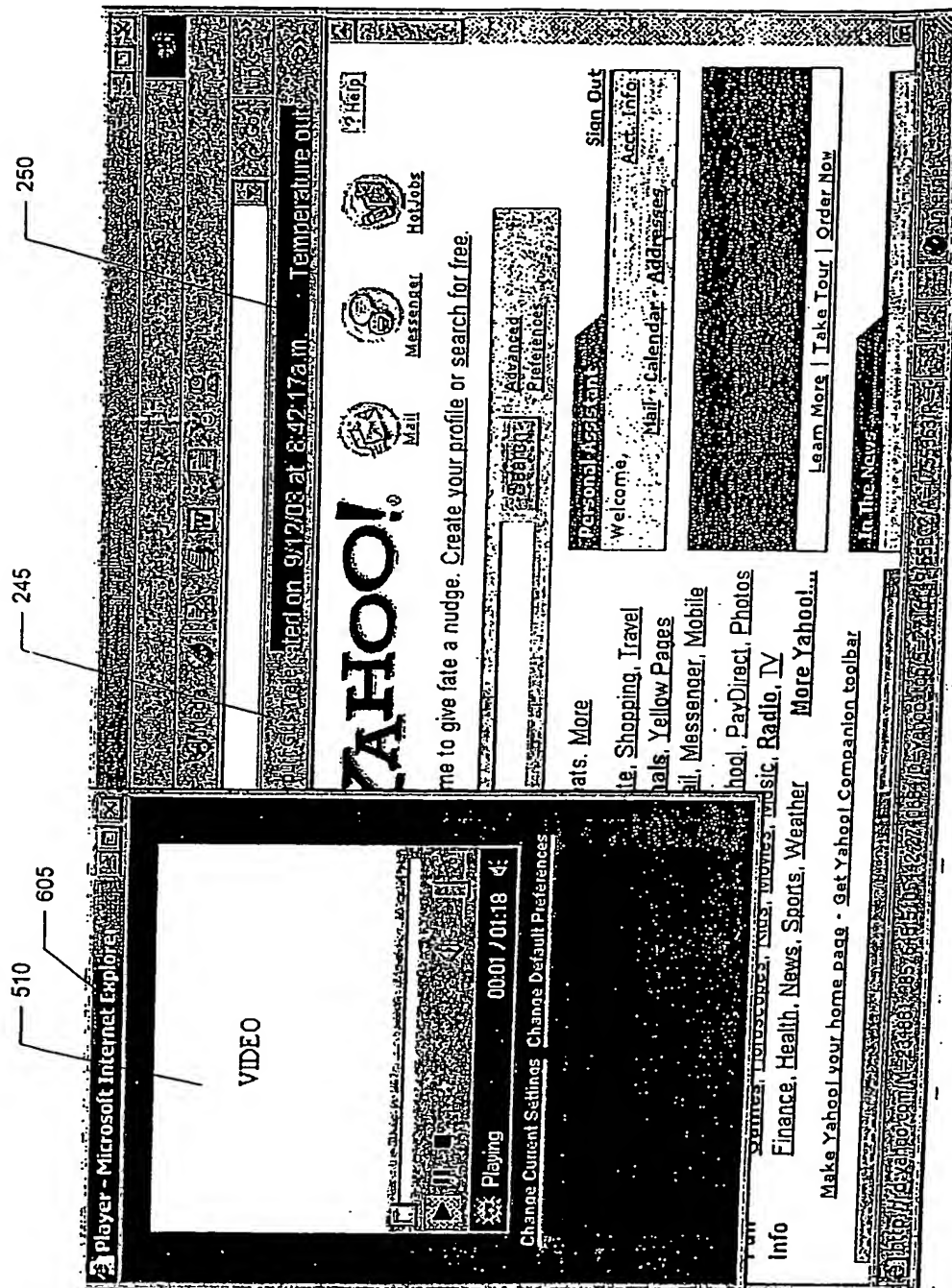


FIG. 6

BEST AVAILABLE COP,

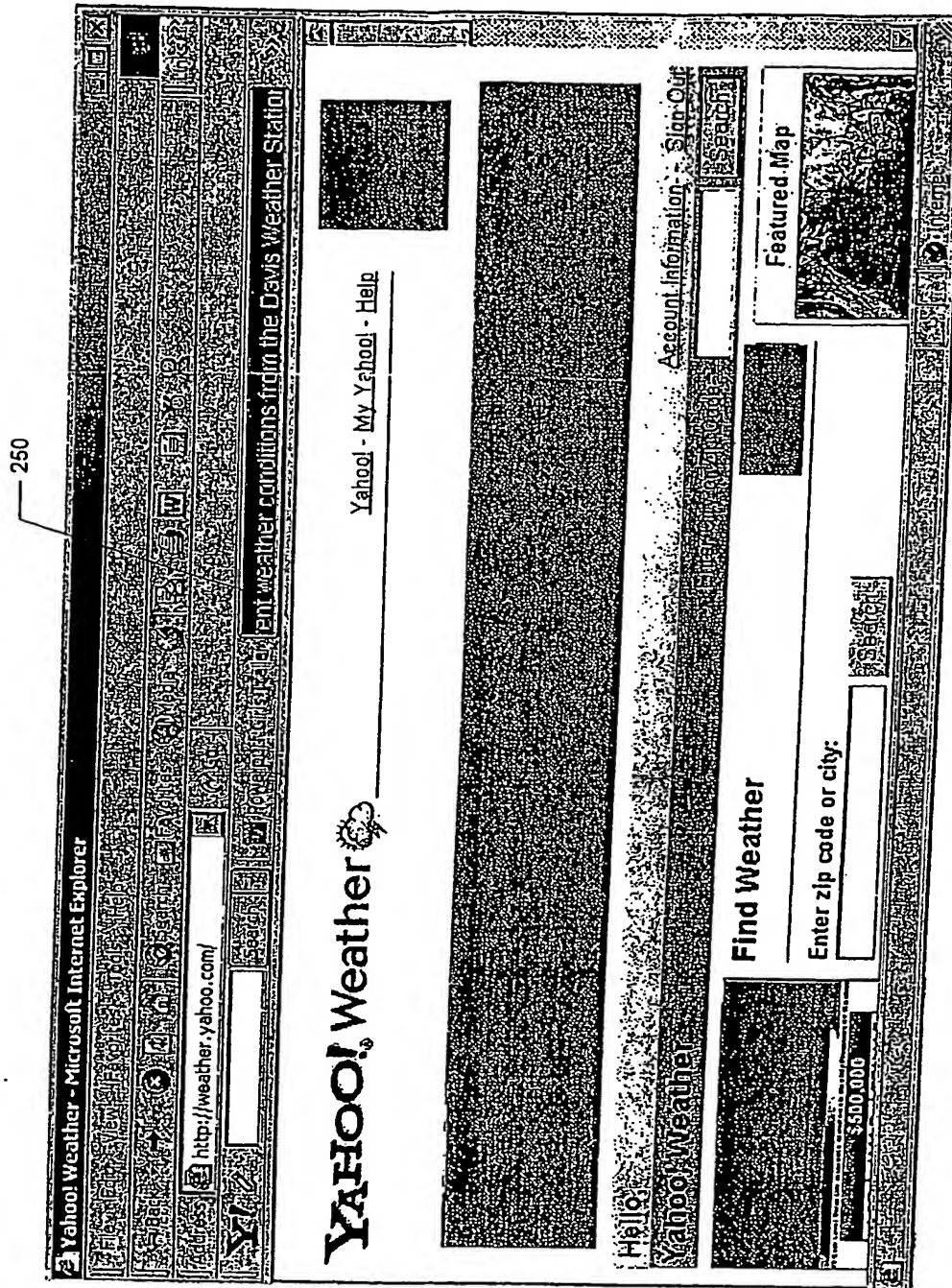


FIG. 7

BEST AVAILABLE COPY

8/10

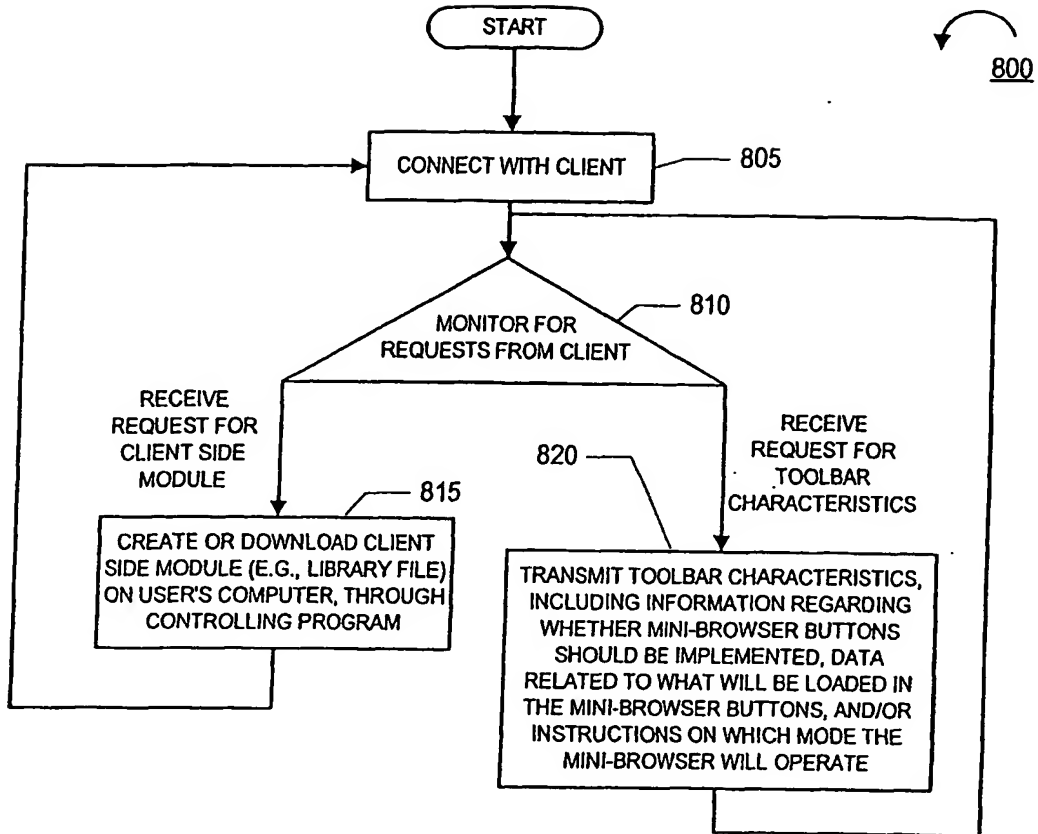


FIG. 8

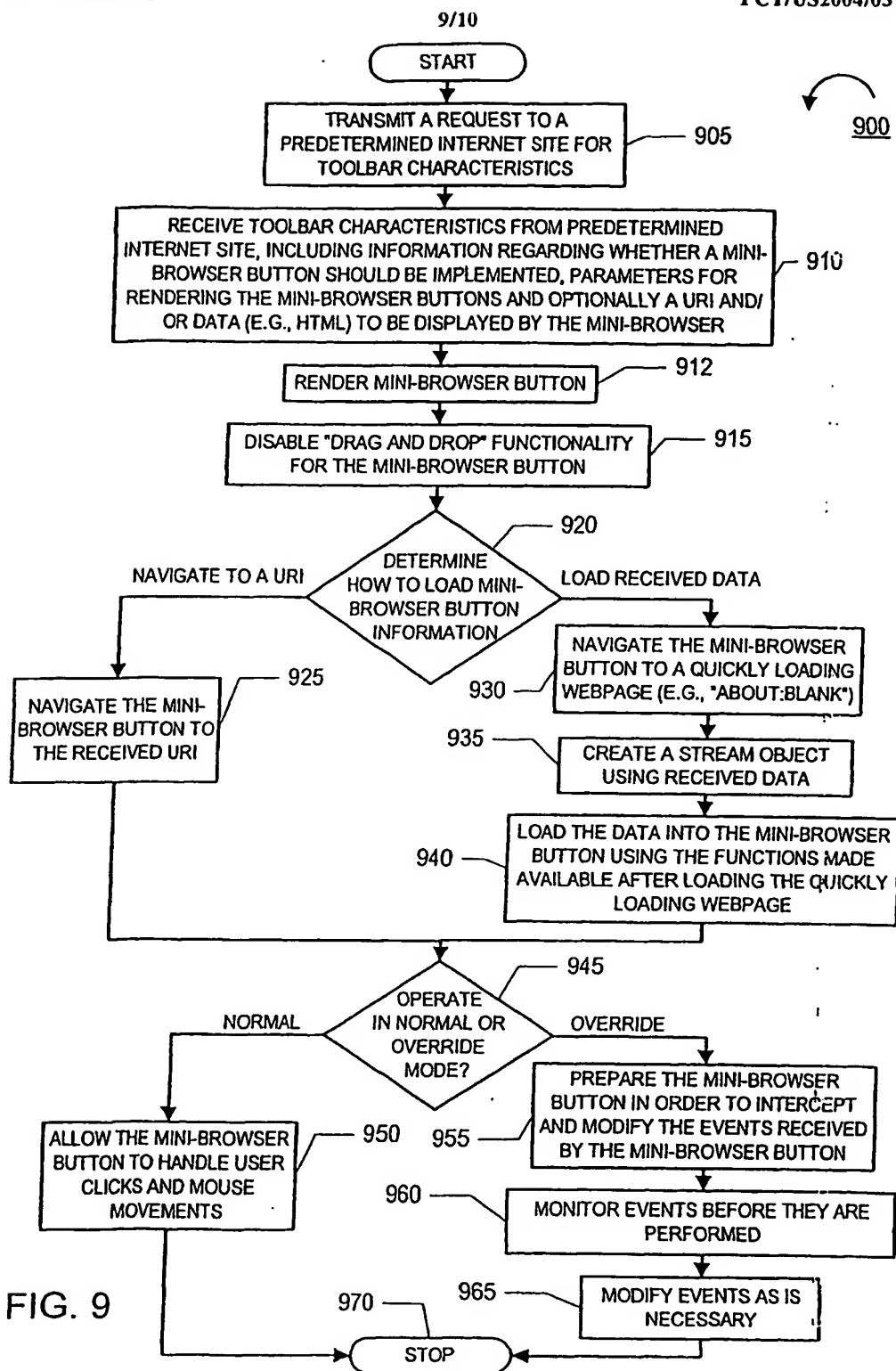


FIG. 9

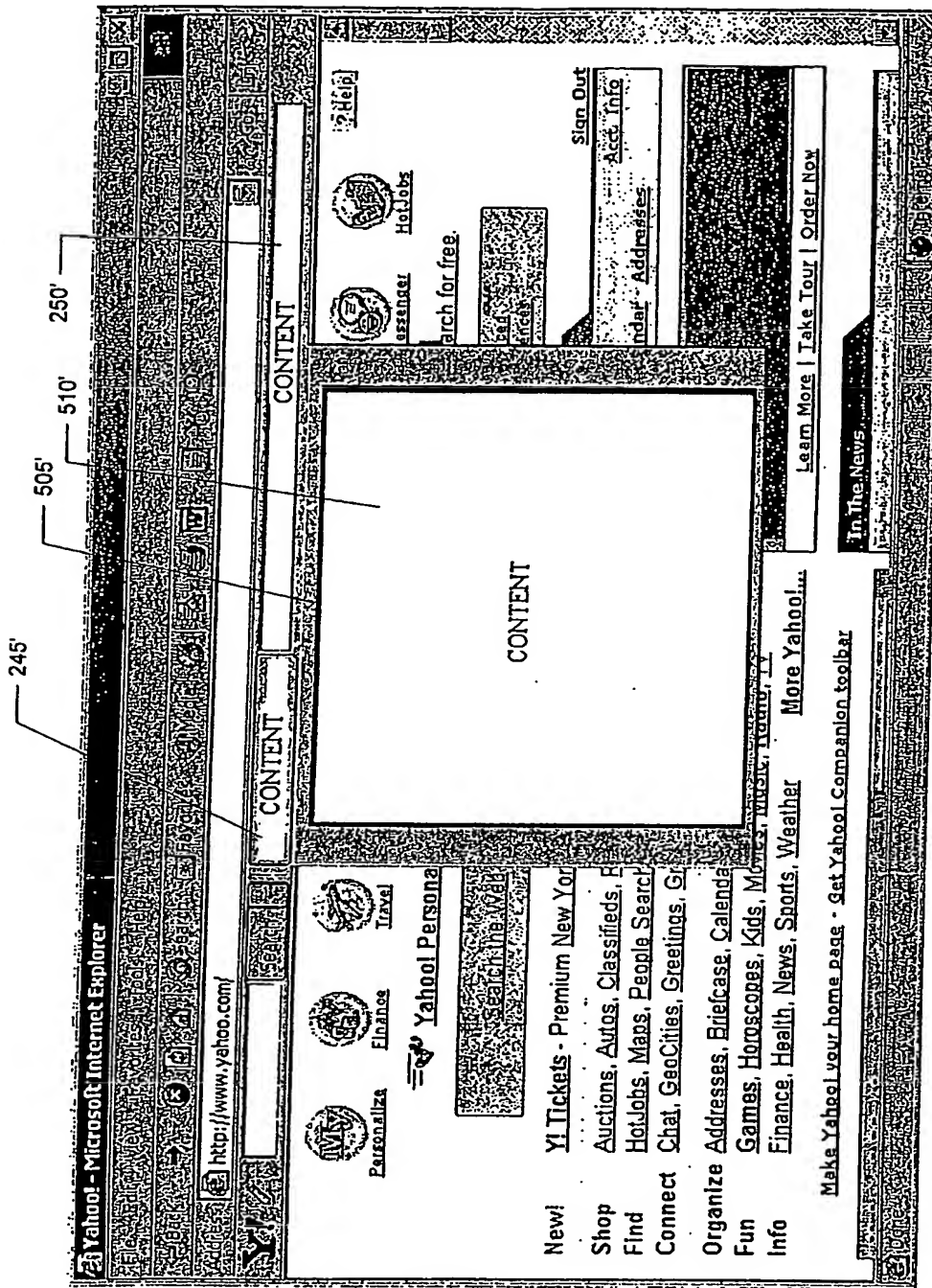


FIG. 10

200

BEST AVAILABLE COPY